

Cordova 插件-iOS

文档目的： Cordova 插件的开发参考。包含的内容有：插件的创建与调试，添加到项目与卸载卸载插件，上传发布，配置权限，版本管理。推荐使用 Xcode IDE 配置权限、打包等，终端命令行打包的形式，若出错不易处理。

1. 开发环境

macOS : 10.13.4

Xcode : 9.3

npm -v : 5.6.0

node -v : 10.1.0

cordova -v : 8.0.0

ionic -v : 3.20.0

plugman -v : 2.0.0

js 编辑器或 IDE

Visual Studio Code/Sublime Text 3 等.

1.1 软件安装:

Xcode 与 Command Line Tools:

网页下载安装: <https://developer.apple.com/download/>

Node.js & npm:

网页下载: <https://nodejs.org/en/>, 升级到最新: `npm install npm@latest -g`

Download for macOS (x64)

8.11.2 LTS

Recommended For Most Users

10.3.0 Current

Latest Features

[Other Downloads](#) | [Changelog](#) | [API Docs](#)

[Other Downloads](#) | [Changelog](#) | [API Docs](#)

安装成功:

```
[➔ ~ node -v
v10.1.0
[➔ ~ npm -v
5.6.0
```

查看对应的版本号

```
➔ MyApp npm version
{ helloworld: '2.0.0',
  npm: '5.6.0',
  ares: '1.14.0',
  cldr: '33.0',
  http_parser: '2.8.0',
  icu: '61.1',
  modules: '64',
  napi: '3',
  nghttp2: '1.29.0',
  node: '10.1.0',
  openssl: '1.1.0h',
  tz: '2018c',
  unicode: '10.0',
  uv: '1.20.2',
  v8: '6.6.346.27-node.6',
  zlib: '1.2.11' }
```

Cordova & ionic Cordova:

安装 Cordova 和 ionic:

```
➔ ~ npm install -g cordova ionic
```

安装成功:

```
[➔ MyApp ionic -v
3.20.0
[➔ MyApp cordova -v
8.0.0
```

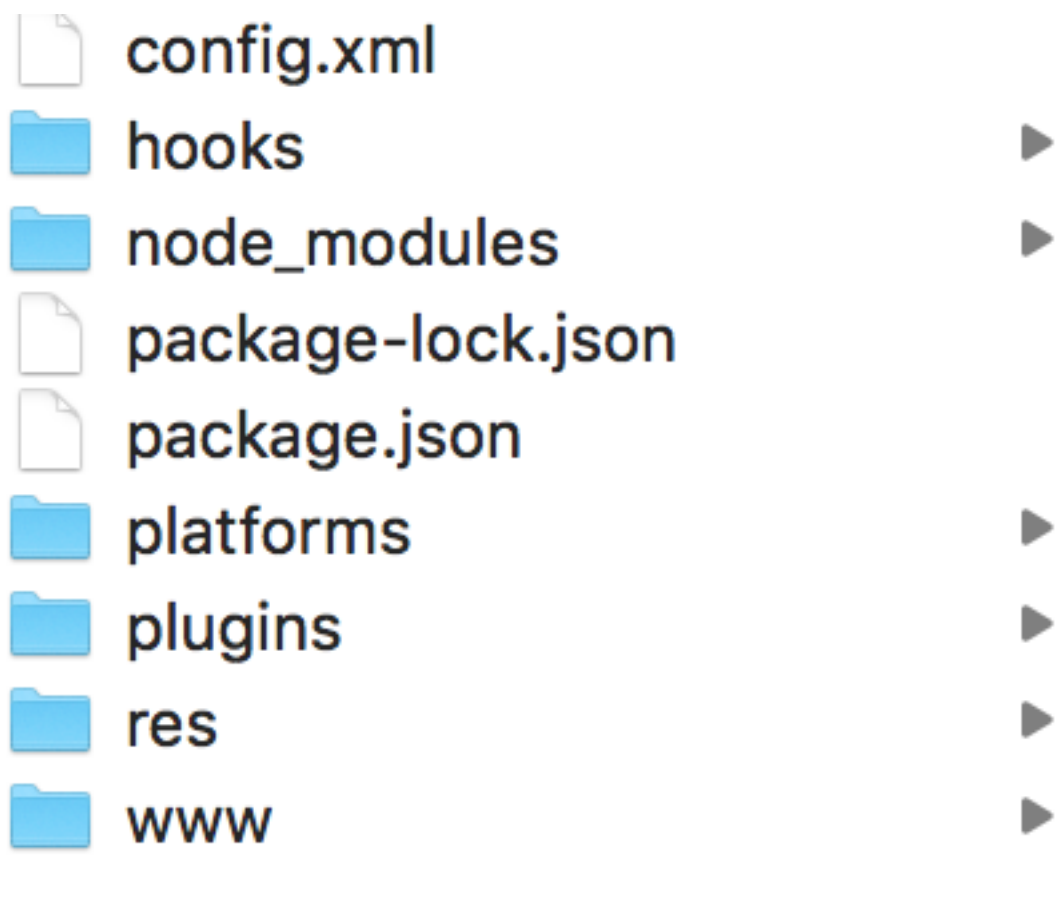
create a project 流程:

```
1.app create
sudo npm install -g cordova (必要,全局仅一次即可)
cordova create myApp com.myCompanymyApp myApp (必要, 创建myApp cordova项目)
cd myApp (必要, myApp项目文件夹下, 即项目根目录)
cordova plugin add org.apache.cordova.file-transfer (无必要, 添加Cordova或者自定义的插件)
cordova platform add ios --save (必要, 添加iOS 平台)
cordova requirements ios (必要, 检查依赖项)

cordova build ios --verbose //构建打包项目(无必要)
cordova run ios //运行iOS代码(无必要)
cordova emulate ios //模拟器运行(无必要)
```

```
[➔ ~ cd Desktop
[➔ Desktop cordova create MyApp
cdCreating a new cordova project.
^R
[➔ Desktop cd MyApp
[➔ MyApp ls
config.xml  package.json  plugins  www
hooks      platforms  res
[➔ MyApp cordova platform add ios
Using cordova-fetch for cordova-ios@~4.5.4
Adding ios project...
Creating Cordova project for the iOS platform:
  Path: platforms/ios
  Package: io.cordova.hellocordova
  Name: HelloCordova
iOS project created with cordova-ios@4.5.4
Discovered plugin "cordova-plugin-whitelist" in config.xml. Adding it to the project
Installing "cordova-plugin-whitelist" for ios
Adding cordova-plugin-whitelist to package.json
Saved plugin info for "cordova-plugin-whitelist" to config.xml
--save flag or autosave detected
Saving ios@~4.5.4 into config.xml file ...
➔ MyApp
```

创建完成, 工程文件夹下应包含



plugman 安装:

```
sudo npm install -g plugman
```

2.开始

Tips:插件的开发, 不涉及工程的创建.

2.1 插件创建的两种方式

项目分为命令行项目即 Cordova 项目(简称 Cordova/Cordova+Pods 项目)和 Cocoapods 集成 Cordova 项目(简称 Pods/Pods+Cordova 项目)、已有项目后期集成 Cordova 项目(简称集成/已有项目)。

2.1.1 终端命令行创建插件

打开终端

菜单栏-- (右上角)--输入 'terminal' 回车, 即可打开终端

或者, Launchpad--其它--终端, 点击即可打开终端

指定存放插件路径文件夹, 如桌面位置:cd Desktop

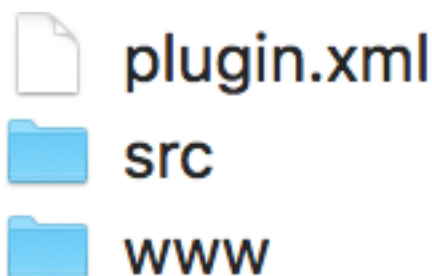
安装 plugman, 终端输入:plugman -v, 若没安装, 则安装升级 plugman,
终端输入:npm install -g plugman

创建插件'`plugman create --name <pluginName> --plugin_id <pluginID> -`
`-plugin_version <version> [--path <directory>] [--variable NAME=VALUE]`', 拆
分为以下几步：

1.创建命令为 `plugman create --name <pluginName> --plugin_id`
`<pluginID> --plugin_version <pluginVersion>`。终端输入:`plugman create --`
`name MyCordovaExamplePlugin --plugin_id cordova-plugin-`
`mycordovaexampleplugin --plugin_version 0.0.1`, 然后回车；

2.到该插件文件夹下'`cd <pluginName>`',终端输入：`cd`
`MyCordovaExamplePlugin`,然后回车；

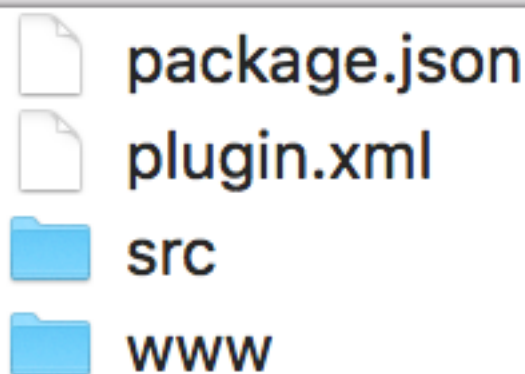
3.添加插件平台 '`plugman platform add --platform_name ios`'，终端输
入：`plugman platform add --platform_name ios`，然后回车；目录如下



4.copy 源文件，在上一步中即可查看 `src` 下出现的 `ios` 文件夹，将
Objective-C 或 swift 源文件拷贝到 `src` 文件夹下。而在实际项目中，按需求随时
新增一些实例方法，为了快速方便调试，就会在项目中调试代码，结果就忘记
了更新该目录下的文件，此时要记得更新该文件夹下的内容，将调试成功无
bug 的源代码 copy 到该文件夹下进行替换。



5.在插件根目录下创建 `package.json` 文件 "`plugman createpackagejson .`"
终端输入：“`plugman createpackagejson .`”，然后回车，可一直回车直到输入
'y'，按默认配置；



6.plugin.xml 的配置

id：唯一标识，无需修改，即为默认；

version：建议 debug 时无需修改，release 时修改一次；

name：无需修改，即为默认；

js-module：无需修改，即为默认("The js-module tag specifies the path to the common JavaScript interface");

clobbers：需要修改,个人理解为马甲，与 JS 中接口/类一致。添加到项目中会根据 xml 生成对应的代码。将 ‘<clobbers target="cordova.plugins.MyCordovaExamplePlugin" />’ 修改为 ‘<clobbers target="MyCordovaExamplePluginClass" />’，在 js 中定义接口为 ‘MyCordovaExamplePluginClass’，才能调用成功。

```
<name>MyCordovaExamplePlugin</name>
<js-module name="MyCordovaExamplePlugin" src="www/MyCordovaExamplePlugin.js">
-   <clobbers target="cordova.plugins.MyCordovaExamplePlugin" /> 默认，不方便使用， -->
    <clobbers target="MyCordovaExamplePluginClass" /><!-- 作为上一步的简化 -->
</js-module>

<engines>
  <engine name="cordova" version=">=7.1.0"/>
  <engine name="cordova-ios" version=">=4.5.0"/>
</engines>
<platform name="ios">
  <config-file parent="/*" target="config.xml">
    <feature name="MyCordovaExamplePlugin">
      <param name="ios-package" value="MyCordovaExamplePlugin" />
    </feature>
  </config-file>
  <source-file src="src/ios/MyCordovaExamplePlugin.m" />
-   引入cocoa pods来管理 -->
-   <framework src="AFNetworking" spec="~> 2.0" weak="false" type="podspec"/> -->
  <framework src="Toast" type="podspec" spec="~> 4.0.0"/>
  <framework src="CRTtoast" type="podspec" spec="~> 0.0.9"/>
  <framework src="SVProgressHUD" type="podspec" spec="~> 2.2.5"/>
```

<!-- 插件源文件 -->

```
<header-file src="src/ios/SSOLoginPlugin.h" />
```

```
<source-file src="src/ios/SSOLoginPlugin.m" />
```

```
<header-file src="src/ios/libs/FHSSO.h" />
```

```
<header-file src="src/ios/libs/MobileArkAgent.h" />
```

```
<header-file src="src/ios/libs/MobileArkPSA.h" />
```

```
<source-file src="src/ios/libs/MobileArk-sdk-ios-4.5.5-build.a" framework="true" />
```

```
<!-- sdk 图片资源 -->
```

```
<resource-file src="src/ios/MAResource/mobark_more_circle@2x.png" />
```

```
<resource-file src="src/ios/SVProgressHUD/SVProgressHUD.bundle" />
```

分割线 -->

<!-- 插件依赖的系统框架 -->

```
<framework src="UIKit.framework" />
```

```
<framework src="Foundation.framework" />
```

```
<!-- info.plist配置 -->
```

```
<config-file platform="ios" target="*-Info.plist" parent="UIBackgroundModes"><!-- 后台 -->
```

```
<array>
```

```
<string>location</string>
```

```
</array>
```

```
</config-file>
```

```
<config-file target="*/Entitlements-Debug.plist" parent="com.apple.developer.in-networking-networkextension">
```

```
<string>app-proxy-provider</string>
```

```
<string>content-filter-provider</string>
```

```
<string>dns-proxy</string>
```

```
</config-file>
```

```
<config-file target="*/Entitlements-Debug.plist" parent="com.apple.developer.in-networking-networkextension">
```

```
<array>
```

```
<string>app-proxy-provider</string>
```

```
<string>content-filter-provider</string>
```

```
<string>dns-proxy</string>
```

```
</array>
```

```
</config-file>
```

7.<pluginName>.js 文件的编程

接口要与 clobbers 一致。如

```
// 7.1
var MyCordovaExamplePluginClass = {
  coolMethodMapper:function (arg0) {
    /* body... */
    exec(success, error, 'MyCordovaExamplePlugin', 'coolMethod', [arg0]);
  },
  showHUDMapper:function (arg0, success, error) {
    /* body... */
    exec(success, error, 'MyCordovaExamplePlugin', 'showHUDMethod', [arg0]);
  }
};
module.exports = MyCordovaExamplePluginClass;
```

在 html 交互时使用

index.html 中,添加调用 js 方法

```
<script type="text/javascript">
    function testShow () {
        // body...
        MyCordovaExamplePluginClass.showHUDMapper('这是传递的值', null, null);
    }
</script>
```

依赖文件的添加

原生文件

.h : <header-file src="src/ios/MyCordovaExamplePlugin.h" target-dir="自己指定映射位置,可不设置"/>

.m : <source-file src="src/ios/MyCordovaExamplePlugin.m" />
<source-file src="src/ios/RegexKitLite/RegexKitLite.m" compiler-flags="-fno-objc-arc"/>

系统库 : <framework src="libbz2.1.0.tbd" />
<framework src="CoreGraphics.framework"/>

第三方库

本地静态库

.a : <source-file src="src/ios/LibSDK/libforSdk.a" framework="true" />
.framework :

<framework src="src/ios/LibSDK/mySDK.framework" custom="true" />

Github 开源库 (本地添加或引入 pods 管理, 建议在项目进行添加)

<framework src="SVProgressHUD" type="podspec" spec="~> 2.2.5"/>

(不建议在此添加, 可能会冲突/重复导入); 然后在 plugin.xml 中配置


```
<engines>
    <engine name="cordova" version=">=7.1.0"/>
    <engine name="cordova-ios" version=">=4.5.0"/>
</engines>
```

, 在 package.json 配置依赖项


```
"engines": {
  "cordovaDependencies": {
    "<2.0.0": {
      "cordova-ios": ">=4.3.0",
      "cordova": ">=6.4.0"
    },
    "2.0.0": {
      "cordova-ios": ">=4.4.0",
      "cordova": ">=7.0.0"
    },
    "<2.1.2": {
      "cordova-ios": ">=4.4.0",
      "cordova": ">=7.1.0"
    },
    "<2.2.0": {
      "cordova-ios": ">=4.5.0",
      "cordova": ">=7.1.0"
    },
    "2.2.0": {
      "cordova-ios": ">=4.5.0",
      "cordova": ">=7.1.0"
    }
  }
},
"dependencies": {
  "babel-plugin-add-header-comment": "^1.0.3",
  "install": "^0.8.2"
}
```

最后，在 platform/ios 中，需要在终端运行 pod install。

```
myCordovaE...$ cd /Users/xingqizhen/Desktop/cordova插件教程/myCordovaE...
Example/platforms/ios
→ ios vim Podfile
→ ios pod install
```

最后打开项目  myCordovaE....xcworkspace 即可。

拓展文件

PacketTunnel.appex (建议放在 git 上，一般需要开启对应的权限、重签名文件) <resource-file src="src/ios/LibSDK/PacketTunnel.appex" />(不建议这样写，

非要这样写, 需要 hook 或者脚本)

资源文件

png 图片 : <resource-file src="src/ios/pngs/yw_1122.png" />

plist : <resource-file src="src/ios/plst/macro.plist" />

bundle : <resource-file src="src/ios/LibSDK/libforSdk.bundle" />

合并(meres)、替换(replace) : 常用于资源文件如 png\plist 等, 适用于 config-file 标签

build setting 配置

请在项目中配置(如果想通过 xml 定义, 推荐查看'cordova-custom-config-master')

.plist 文件设置(如果较多, 推荐在项目中配置)

```
<config-file platform="ios" target="*-Info.plist"
parent="NSAppTransportSecurity"><!-- ATS -->
  <dict>
    <key>NSAllowsArbitraryLoads</key>
    <true/>
  </dict>
</config-file>

<config-file platform="ios" target="*-Info.plist"
parent="LSApplicationQueriesSchemes"><!-- 白 名 单
mode="merge",mode="replace",mode="delete" -->
  <array>
    <string>com.hoperun.sharecar</string>
    <string>com.myDemo.sso2</string>
  </array>
</config-file>
```

.entitlements 文件(如果较多, 推荐在项目中配置)

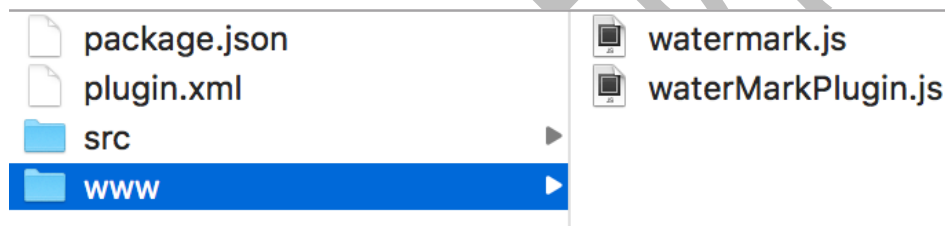
```
<config-file platform="ios" target="*/Entitlements-Debug.plist"
parent="keychain-access-groups">
  <array>
```

```
<string>$(AppldentifierPrefix)com.Code.SSO</string>
<string>$(AppldentifierPrefix)com.hoperun.sharecar</string>
</array>
</config-file>
```

```
<config-file platform="ios" target="*/Entitlements-Release.plist"
parent="keychain-access-groups">
  <array>
    <string>$(AppldentifierPrefix)com.Code.SSO</string>
    <string>$(AppldentifierPrefix)com.hoperun.sharecar</string>
  </array>
</config-file>
```

多.js 文件，有时需要导入其它 js 文件:

```
<asset src="www/watermark.js" target="plugins/cordova-plugin-waterMark/www/watermark.js"/>
```



6.JS 编写

特别需要注意的是，接口需要与 'clobbers' 中的一致，

```
<clobbers target="MyCordovaExamplePluginClass" /><
```

```
module.exports = MyCordovaExamplePluginClass;
```

7.其它注意事项

插件名命名规则，不要用空格；

插件 id 命名规则，类似 bundle identity 域名反转如 com.company.pluginName 或者遵循官网格式如 cordova-plugin-camera；

在 plugin.xml 配置完后，就无需在 JS 中再做额外配置，系统权限则需要额外进行配置。

2.1.2 已有 Cordova 项目中创建/添加插件并调试

1.打开工程，新建文件，作为 CDVPlugin 的子类

2.修改.h 文件, #import <Cordova/CDV.h> , 并添加自定义方法

3.在.m 中实现该方法

4.在 config.xml 中进行配置

```
'<feature name="myExamplePlugin">
    <param name="ios-package" value="myExamplePlugin" />
</feature>'
```

5.js 文件的编写, 参考 2.1.1 中 js 文件的编写规则

6.如果是在已有项目中, 集成了 Cordova, 则 cordova_plugins.js 文件无需额外配置

7.在 html 中进行调用, 通过调用 js 接口并调试。如在 html 中加入

```
<script type="text/javascript">
    function testShow () {
        // body...
        MyCordovaExamplePluginClass.showHUDMapper('这是传递的值', null, null);
    }
</script>
```

即可调试

8.Objective-C/Swift 源文件的编写

```
#import <Cordova/Cordova.h>
#import <Cordova/CDV.h>
@interface myExamplePlugin : CDVPlugin
```

```
- (void)showHUD:(CDVInvokedUrlCommand *)command;
```

```
@end
```

```
- (void)showHUD:(CDVInvokedUrlCommand *)command{
    NSString *backId =command.callbackId;
    NSString *param1 = [command.arguments objectAtIndex:0];
    NSString *param2 = [command.arguments objectAtIndex:1];
    NSString *param3 = [command.arguments objectAtIndex:2];

    CDVPluginResult *result = nil;
    if (command.arguments) {
        //进行代码编写

        result = [CDVPluginResult resultWithStatus:CDVCommandStatus_OK];
    }else{
        //进行代码的编写
        result = [CDVPluginResult resultWithStatus:CDVCommandStatus_ERROR];
    }
    [self.commandDelegate sendPluginResult:result callbackId:backId];
    [self.commandDelegate evalJs:@"调用JS代码"];
    dispatch_async(dispatch_get_global_queue(0, 0), ^{
        //主线程执行操作
    });

    [self.commandDelegate runInBackground:^(
        //并发全局队列执行操作
    )];
}
```

2.1.3 不包含 Cordova 的项目创建

需要先将 Cordova 集成到项目中去，其它与 2.1.2 一样

2.2.1 插件调试

需要添加后再进行调试，JS 部分依据 js 调试方法即可。参考 2.1.2

2.3.1 添加插件

Cordova 项目,对应的命令 'cordova plugin add <pluginName|pluginName path|plugin-id>' ;Pods 项目,则在 Podfile 中添加;集成项目,需要手动添加。

Cordova 项目：cd 到工程根目录，终端输入：

cordova plugin ls，然后 Enter 键； cordova plugin save，然后 Enter 键；

cordova plugin add 本地插件文件夹路径/http 插件地址

➔ myCordovaExample cordova plugin add /Users/xingleizhen/Desktop/Cordova插件教程/MyCordovaExamplePlugin

cordova plugin add cordova-plugin-camera --variable
name="CAMERA_USAGE_DESCRIPTION" value="My App would like to access
your camera, to take photos of your documents."

Cocoapods 项目：

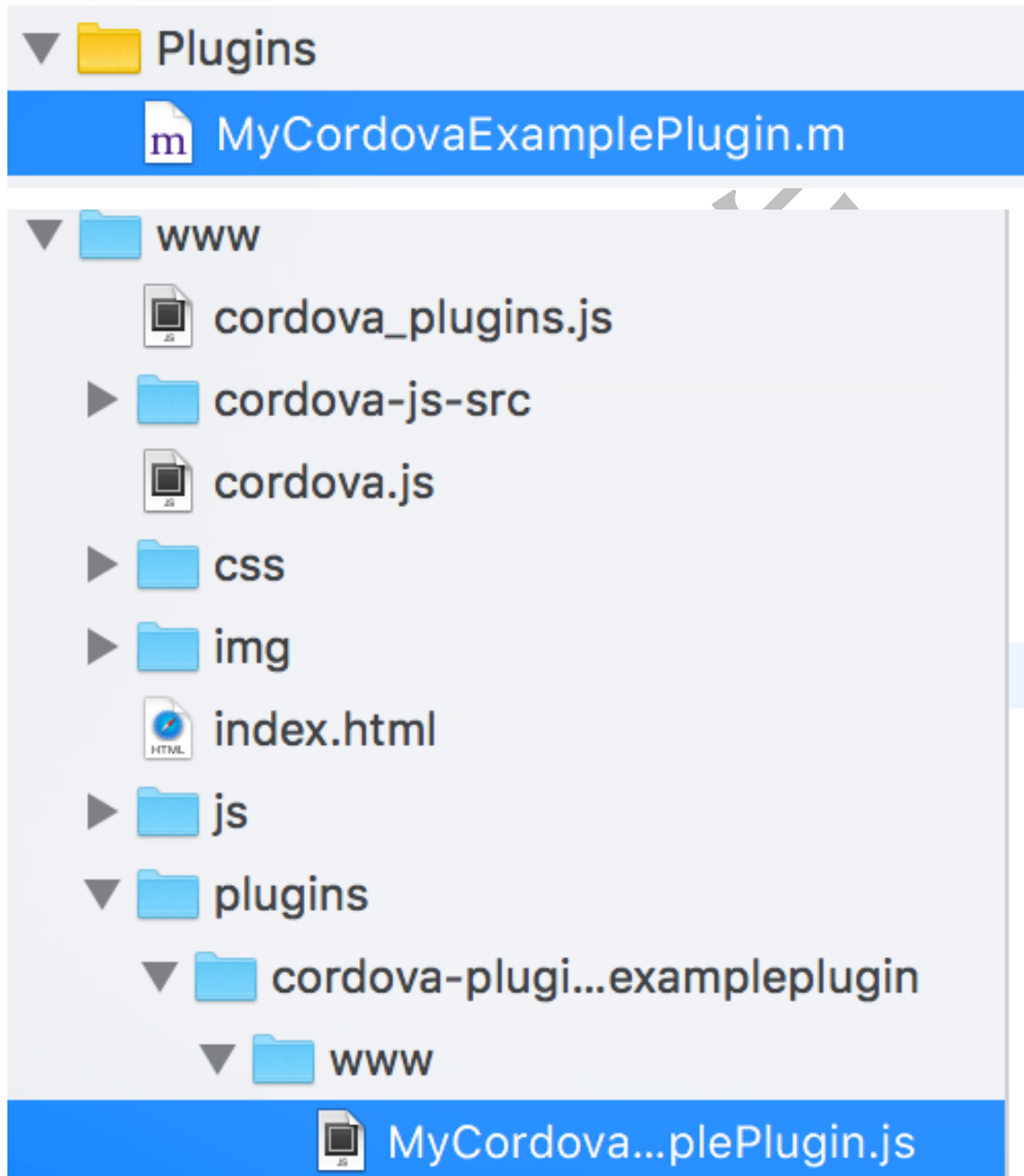
需要在 Podfile 文件中加入

```
pod 'cordova-plugin-camera', '~> 2.3.0'
```

```
pod 'CordovaPlugin-camera', '~> 3.0.0'
```

集成项目：

需要手动导入，拖进去即可，js 放到 www 文件夹下。



2.3.2 卸载插件

```
cordova plugin rm <plugin-id>
```

2.3.3 更新插件

需要先移除插件，再添加

2.4 插件的上传与发布

可上传到 github、npm 上，查看教程即可

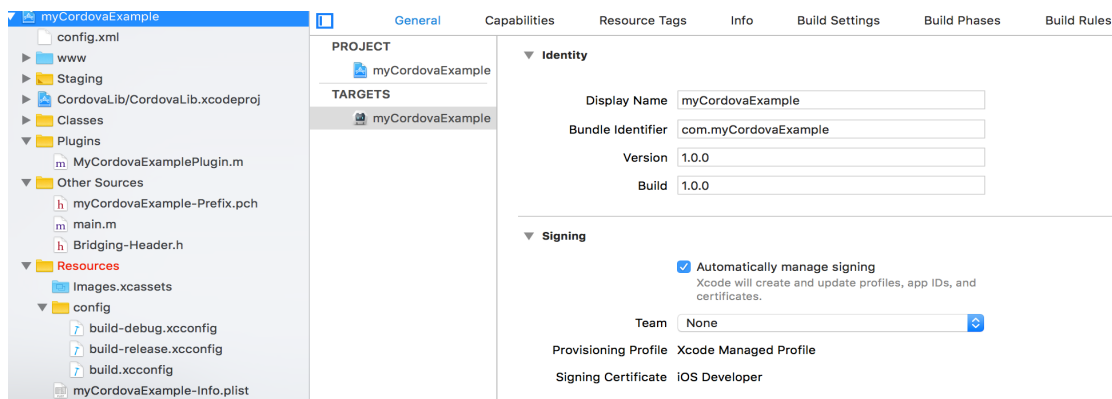
2.5 版本管理

更改 plugin.xml 中的版本号即可。

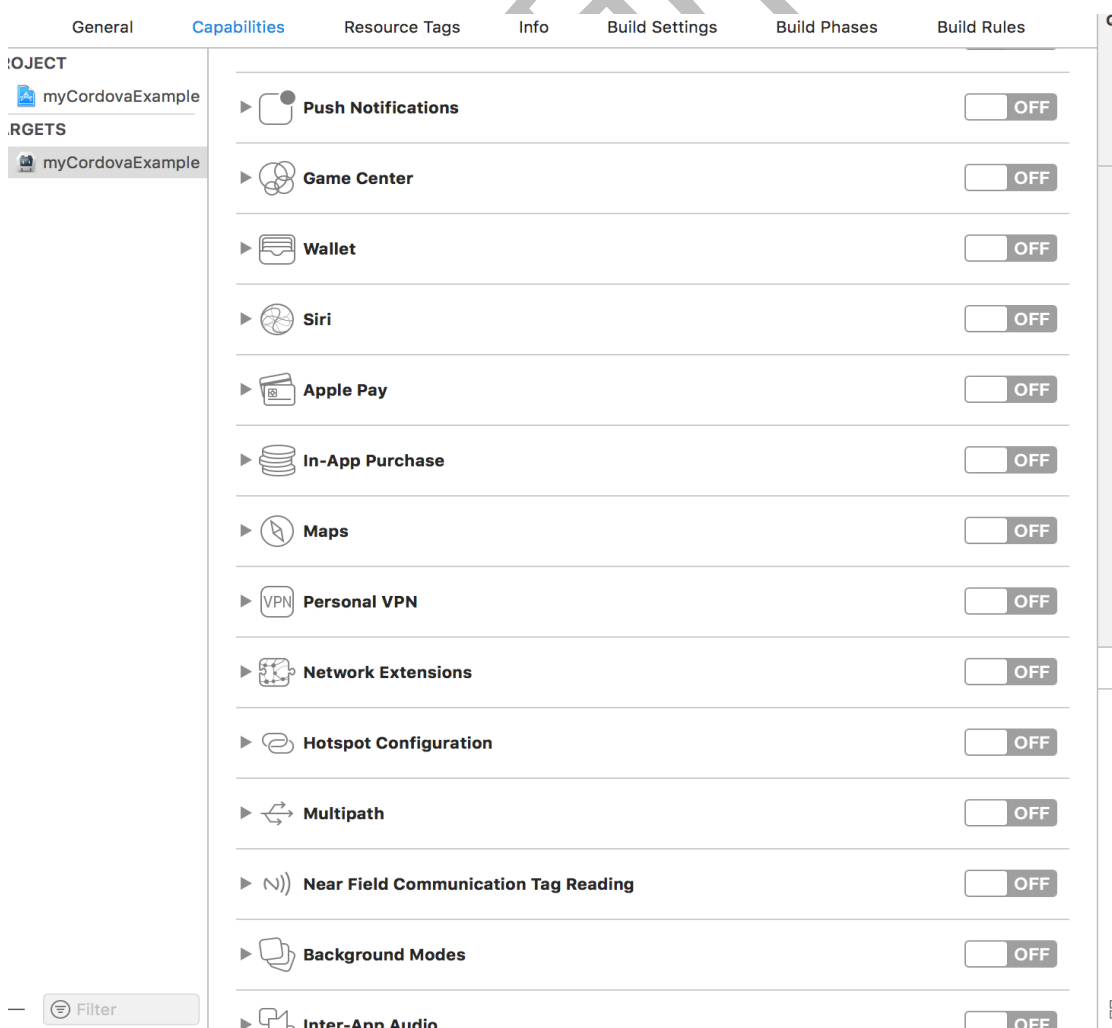
2.6 App 打包

命令行打包不推荐，推荐原生 Xcode 打包，配置权限；

项目基本信息、证书等



开启对应的权限



项目配置，真机/模拟器、link 头文件、证书的配置等等

General

Capabilities

Resource Tags

Info

Build Settings

Build Phases

Build Rules

OBJECT

myCordovaExample

RGENTS

myCordovaExample

Basic

Customized

All

Combined

Levels

+

Q Search

Other OSACompile Flags

Save as Execute-Only

No

Static Analyzer - Analysis Policy

Setting

myCordovaExample

Analyze During 'Build'

No

Mode of Analysis for 'Analyze'

Deep

Mode of Analysis for 'Build'

Shallow (faster)

Static Analyzer - Generic Issues

Setting

myCordovaExample

Dead Stores

Yes

Improper Memory Management

Yes - \$

Misuse of 'nonnull'

Yes

Static Analyzer - Issues - Apple APIs

Setting

myCordovaExample

Improper Handling of CFError and NSError

Yes

Missing Localizability

No

Missing Localization Context Comment

No

Misuse of Collections API

Yes

Misuse of Grand Central Dispatch

Yes

Suspicious Conversions of NSNumber and CFNumberRef

Yes

Static Analyzer - Issues - Objective-C

Setting

myCordovaExample

@synchronized with nil mutex

Yes

Improper Instance Cleanup in '-dealloc'

Yes

Method Signatures Mismatch

Yes

Misuse of Objective-C generics

Yes

Unused Ivars

Yes

Violation of 'self = [super init]' Rule

Yes

Violation of Reference Counting Rules

Yes

Filter

导入系统库、第三方库、源文件等等

Capabilities

Resource Tags

Info

Build Settings

Build Phases

Build Rules

+

Filter

Target Dependencies (1 item)

[CP] Check Pods Manifest.lock

x

Copy www directory

x

Copy Bundle Resources (3 items)

x

Compile Sources (4 items)

x

Link Binary With Libraries (2 items)

x

Name

Status

libCordova.a

Required

libPods-myCordovaExample.a

Required

+

-

Drag to reorder frameworks

[CP] Copy Pods Resources

x

其它注意事项：

Cordova 版本需要与 npm 版本对应，npm 版本可手动降低(问题无法重现).

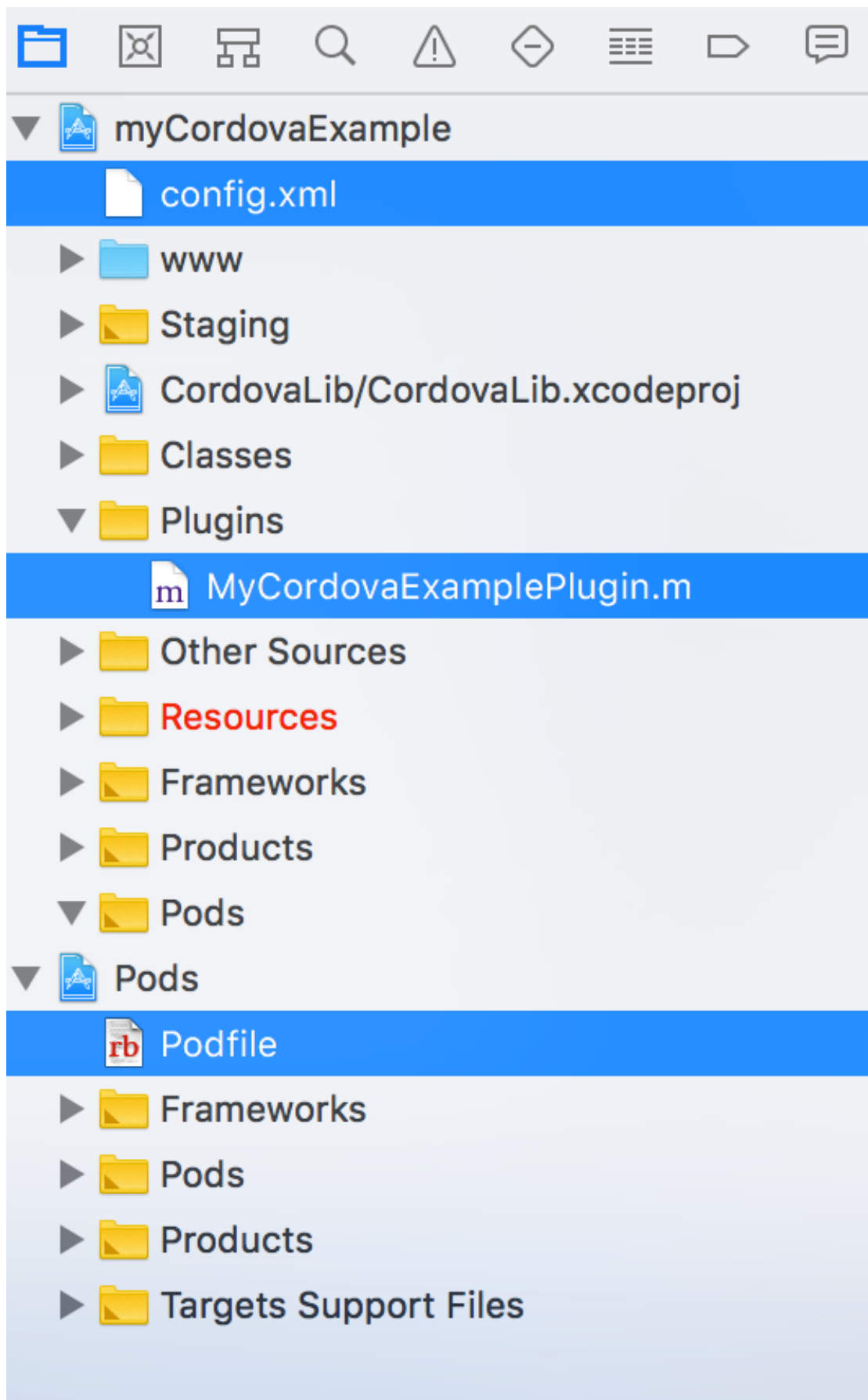
如果出现权限的问题，请命令前加 “sudo” .

Project 中的 Bitcode 看项目和库的具体要求，设置 yes/no.

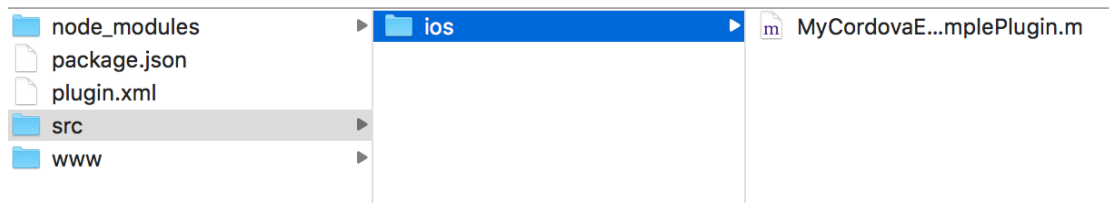
附

示例项目结构

版权所有



插件目录结构



若有疑问，可联系 email:Rayz_17@126.com;
qq:1344957728@qq.com.